A Mixed-Level MOS Logic Simulator
Utilizing a New Continuous Strength Algebra (CSAL)

Jin-Hyeong Kong  Stephen A. Szygenda
Electrical and Computer Engineering Department
The University of Texas at Austin
Room 240 Engineering Science Building
Austin, Texas 78712-1080

Abstract
This research presents a new evaluation method for logical MOS gates. The approach is suitable for mixed-level simulation of gates and switches. A logical MOS gate models a driver-load transistor network, performing a Boolean logic function, in a static manner. The gate is normally represented by a Boolean expression, of which conventional evaluations at the gate-level provide the signal level, but not the signal strength of the gate output. In order to overcome this limitation, a new expression (compatible with the Boolean expression) is defined over a new Continuous Strength Algebra (CSAL), and it is then evaluated to provide the signal level and strength for the gate output. This approach results in achieving the computation speed of gate-level, gained by using the higher level of abstraction; and the accuracy of switch-level, obtained by utilizing the new algebra.

I. Introduction
Most switch-level simulators partition a digital MOS transistor network into several mutually interacting subnetworks. The key to the partitioning strategy is to simplify the complexities of the network operation and to minimize the mutual interactions between the subnetworks. Usually, the partitioned network includes two kinds of subnetworks: one is a bidirectional subnetwork and the other is a unidirectional subnetwork. The bidirectional subnetwork actually requires the overall capability of conventional switch-level simulation [1] to resolve peculiar characteristics of MOS circuits, including; dynamic storage, charge sharing, bidirectional behavior of the transistor, sneak paths, etc. The unidirectional subnetwork is associated with the following features; (1) it has multiple inputs and a single output, (2) it consists of two series-parallel transistor networks, called a Driver-Load configuration, where one forms a driver network between the GND power node and the output node; the other is a load network between the VDD power node and the output node, (3) it performs a Boolean logic function in a static manner. This means that the Boolean logic function is due to the state of conduction, or no conduction, of transistor networks. This kind of subnetwork can be abstracted as a unidirectional logical primitive, which will be called a logical MOS gate. Throughout this paper, the discussions will be outlined and described for n-channel MOS (nMOS) and complementary MOS (CMOS) circuits only. The extensions to handle p-channel (pMOS) circuits can be easily derived.

In this paper, we focus our attention on the evaluation of logical MOS gates. This primitive need not be evaluated at the switch-level, since its internal nodes do not interact with each other and its internal transistors transmit signals, in a fixed direction, to the output node. The conventional switch-level simulator [1] evaluates such gate primitives in a redundant fashion. Since the evaluation is based on mutual interactions between neighboring nodes in channel-graphs of partitioned subnetworks, all nodes within each graph have to be considered and reevaluated for each event. Furthermore, users are not interested in the signals at internal nodes, but only the signal at the output node. In recently proposed mixed-level simulators [2,3,4,5], including switch-level, a unidirectional subnetwork is modeled as a higher logical abstraction of a gate, to reduce the simulation time. In order to maintain the accuracy consistent with switch-level simulation, the gate has to be evaluated with a strength capability. Some simulators [2,3] have associated the output of the gate with a limited discrete strength set, whereby reflecting the output signal level and MOS circuit techniques. However, this limited strength capability restricts the performance of its switch-level simulation for bidirectional subnetworks, and does not suffice for predicting the correct signal strength of the gate output. In other simulators [4,5], based on continuous strength values, one of two equivalent resistances (i.e., pull-up and pull-down, RH/RL) from the gate output is assigned to the strength of the output signal, depending on the output logical state (High or Low). Although these simulators do not limit the strength values and the accuracy of their switch-level simulations, they cannot predict the correct equivalent resistance, for most complex gates having more than one input, with the two resistances. Besides, since the equivalent resistances of RH and RL are normally utilized in computing a single (rising/falling) delay, the erroneous prediction causes inaccuracy in simulating multiple inputs-to-output delays for the gate. This shows basic limitations that are inherent in using a logical gate abstraction in the mixed-level simulation.

In this research, a new evaluation method for logical MOS gates, in mixed-level simulation, is developed based on a new algebra over a continuous strength set. An element in the strength set is represented by a pair (SVDD,SGND), where SVDD and SGND are continuous charging and discharging strength values, respectively. The new algebra is utilized to evaluate a Boolean expression corresponding to the logical MOS gate and to provide the signal strength (SVDD,SGND) for the gate output. Furthermore, its strength values of SVDD and SGND are used to develop a timing model of multiple input-to-output delays, which are varied with the transistor network and its input signals, for logical MOS gates. This timing model can overcome the inherent limitation of logical gate abstraction in the timing simulation; a single (rising/falling) delay.

II. Logical MOS Gate
As mentioned earlier, a logical MOS gate indicates a driver-load transistor network performing a Boolean logic function in a static manner. With the transistor network based on a Boolean description, this gate carries out a Boolean function dictated by the state of conduction, or no conduction, of transistor networks between the power nodes and the output node.

In this section, we describe a method to derive a driver-load transistor network from an arbitrary complementary Boolean expression. This method illustrates a certain correspondence of
the driver-load transistor networks with the Boolean expressions. Then, we discuss various features of two logical MOS gates; a Static Gate and a High Impedance Gate.

A. Derivation of transistor networks

The general procedure to derive a driver-load transistor network for an arbitrary complementary Boolean function \( f \) is as follows. Start with an expression for the Boolean function \( f \), and obtain the driver network of a series-parallel transistor network. In the driver network, an AND operation corresponds to transistors connected in parallel and an OR operation corresponds to transistors connected in series. The derived transistor networks are illustrated in Figure 1.

![Fig. 1 Transistor network vs. Boolean expression](image)

This correspondence, between a driver-load transistor network and a Boolean description, allows us to represent an internal transistor network of a logical MOS gate by using a complementary Boolean expression. That is, a configuration of transistors in the driver-load transistor networks, can be described by a complementary Boolean expression.

B. Static Gate

A static gate is a device with multiple inputs and a single output. This gate implements a composite Boolean function consisting of ANDs, ORs, and a NOT, as shown in Figure 1 (a) or (b). It is described by a complementary Boolean expression or a bubbled-output Boolean gate. The static gate comprises the driver and load networks which are derived from the Boolean expression, as explained above. Figure 2 (a) and (b) show two typical configurations of CMOS and nMOS static gates.

For any combinations of the gate inputs, the static gate always forms transistor conduction paths between the power nodes and the output node. In a CMOS circuit, two possible conduction paths, in the driver and load networks, are mutually exclusively formed to the output node. However, in an nMOS circuit, the load network of a single depletion transistor is always turned-on and the conduction path in the driver network is conditionally formed, depending on the gate inputs. In the case that the output node is electrically connected to both power nodes, its logical behavior is dictated by a ratio of resistances of the two conduction paths in the driver and load networks. Thus, in the static state, the state of conduction or no conduction, which is controlled by the gate inputs in the two networks performs a Boolean logic function at the gate output.

C. High Impedance Gate

When an output-enable function is added to a CMOS static gate, it implements the static gate with an additional output state of Z. A high impedance gate represents a device with multiple inputs, two enable clock inputs, and a single output. The enable clock inputs controls the connectivity of two enhancement transistors serially connected to the load and driver networks, which are derived from a complementary Boolean expression, as show in Figure 2(c). This gate is described by a complementary Boolean expression and the two enable clock inputs.

Depending on the enable clock inputs, the output node can be isolated from the power nodes in a disable state, the output falls into the Z state; and its signal is determined from whether its nodal capacitance was in a charged or discharged state, (i.e., the previous state of the output node). When the high impedance gate is in the enable state, it behaves like a static gate. Then, the signal at the output node is determined from the state of conduction, or no conduction, in the two networks. In the disable state, even though the gate dynamically holds the information at the output node, it is possible to unidirectionally handle the high impedance gate with the additional Z state. The output node does not interact with the rest of the transistor network, in the disable state.

III. Signal Strength

The behavior of MOS circuits is based on resistive and capacitive features, which are due to finite resistance of the on-transistor and capacitance of the node. Most simulators include models which are either discrete [1] or continuous [6], in representing the resistance and the capacitance, and in handling signals at internal nodes. In this research, the demand for greater accuracy leads us to use a continuous model.

With the continuous model, the elements in an MOS circuit are evaluated to yield signals at nodes. A signal is comprised of two entities; level and strength. The signal level represents logical behavior of the circuits by using three logical voltage levels; Low (0), High (1) and Unknown/Invalid (X). The signal strength characterizes impedance, which is modeled by two resistances between the two power nodes (i.e., VDD and GND) and an internal node, or the amount of charge in a nodal capacitance.
When an internal node is electrically connected to the power node(s), a static strength is defined for the node. In order to represent the static strength, two resistances of the node, with respect to the VDD and GND power nodes, are used in pairs (SVDD, SGND). Actually, SVDD and SGND denote possible charging and discharging resistances (or strengths) at the node, respectively. If the node has a conduction path to one of the power nodes, one resistance is finite, r, and the other resistance (SVDD, SGND). Actually, SVDD and SGND denote possible becomes infinite, ∞. The signal level is determined from which resistance is finite; and the signal strength is inversely proportional to the amount of resistance. Especially in nMOS, two conduction paths can be represented by (ri, r2), where r1 and r2 are finite resistances.

To account for its signal level, which power node provides a node is isolated from the power nodes, (SVDD, SGND) used to determine the signal level. In the case that an internal node is isolated from the power nodes, (SVDD, SGND) becomes infinite, ∞. Then, a dynamic strength is defined as a charge amount in the nodal capacitance, which is dictated by the previous signal level at the node.

In order to formulate the strength (or resistance of an internal node, we need to model the connectivity of an MOS transistor, in the transistor networks. Corresponding to the signal levels of 0, 1 and X at the transistor-gate terminal, the connectivity of an MOS transistor is classified into one of three states: On, Off, and Unknown. An off-transistor is modeled as an infinite resistor ×, and an on-transistor is modeled as a finite resistor r. Actually, the size of r is primarily dependent on the geometry (W/L) of the transistor. In the case of the unknown connectivity, the transistor is characterized by an unknown resistor x, which can have any value of a finite resistor r, or infinity ∞.

IV. Algebraic Formulation

In order to deal with the signal level, a well-defined Boolean algebra had been developed. However, this algebra does not suffice in handling MOS features, for which a strength capability is required. As a further step (toward evaluation for logical MOS gates), we will describe a new algebra to handle Boolean expressions with strength capability. This new algebra is designed to compute a signal strength for a Boolean expression, which can be translated into the same signal level (i.e., O/l/X) as those of the Boolean algebra. An continuous strength Algebra (CSAL).

Suppose that S is a set. S consists of elements represented by a pair (RVDD, RGND), in which RVDD (or RGND) is a possible charging (discharging) strength value. The strength value is a real number taking its value from [0,∞] or a symbol representing an unknown value in the interval [0,∞];

S = {(RVDD, RGND); RVDD, RGND ∈ U}

where U = [0,∞] U {x}; an element of U is either 0, an arbitrary real number r ∈ (0,∞), ∞, or the symbol of unknown value x. Since the strength values (i.e., RVDD and RGND) can have any real number from [0,∞], the set S will be called a continuous strength set.

Let A = (a, b) and B = (c, d), where a, b, c, d ∈ U, be elements of S. Then, algebraic operations are defined as follows:

\[
\begin{align*}
\text{OPAND}(A, B) &= (a + c, l/(1/(b + 1/c))) = (a + c, b[c, b + d]) \\
\text{OPOR}(A, B) &= a + B = (a, b + (c, d)) = (l/(l/a + 1/c), b + d) = (a[c, b + d]) \\
\text{OPINV}(A) &= A = (-a, b) = (b, a) \quad \text{where a notation of } 1 \text{ will be called a parallel addition.}
\end{align*}
\]

There are two arithmetic operations of addition + and parallel addition || for the elements in U. The computation rules of the arithmetic operations are shown in Table 1. The OPAND operation • yields the addition of the charging strengths and the parallel addition of the discharging strengths. The OPOR operation + yields the parallel addition of the charging strengths and the addition of the discharging strengths. The OPINV operation yields the interchange of the two strength values in a pair. The precedence of operations in the set S exactly corresponds to that in the Boolean algebra, namely: (1) Parenthesis, (2) OPINV (INV), (3) OPAND (AND), (4) OPOR (OR).

Since these algebraic operations are defined on the continuous strength set S, we will call it a Continuous Strength Algebra (CSAL).

The CSAL is involved with the following features:

(1) The set S is closed with respect to the two binary operations of • and + and the unary operation of ;

S X S → S

(2) The operations of • and + are associative;

\[
\begin{align*}
(A + B) + C &= A + (B + C), \\
(A \bullet B) \bullet C &= A \bullet (B \bullet C),
\end{align*}
\]

(3) The operations of • and + are commutative;

\[
\begin{align*}
A \bullet B &= B \bullet A, \\
A + B &= B + A
\end{align*}
\]

(4) (0,∞) is the identity element for • and the annihilator for ;

\[
\begin{align*}
(0,\infty) \bullet A &= A, \\
(0,\infty) + A &= (a, 0), \quad \text{for } A \in S
\end{align*}
\]

(5) (0,∞) is the identity element for + and the annihilator for •;

\[
\begin{align*}
(0,\infty) + A &= A, \\
(0,\infty) \bullet A &= (0,0), \quad \text{for } A \in S
\end{align*}
\]

(6) The operation of • obeys the inversion law,

\[
\neg(A \bullet A) = A \quad \text{for } A \in S
\]

(7) De Morgan’s theorem [7] in the set of S;

\[
\neg(A_1, A_2, A_3, ..., A_n, \bullet, +) = \neg(A_1, +, A_2, A_3, ..., \bullet, +)
\]

for Ai ∈ S, i = 1, ..., n
B. Mappings between Level and Strength

Normally, the Boolean set B consists of 0, 1, and X elements; and the algebra is defined over the B. Mappings between Level and Strength limitations in handling peculiar Unknown, respectively. The 0.1 and X elements in the set resistive and capacitive behaviors. The limitation can be derive the signal strength from a Boolean expression. The operations of OPAND, OPOR and OPINV in the set CSAL, two mapping relations, between the set B and the set S, have to be defined, namely; I: B→S and O: S→B, which are shown in Figure 3. The mapping of I: B→S indicates that the signal level of an input i of the Boolean expression is encoded in an element, consisting of two strength values, in the set S, which is defined as follows:

\[ 0 \rightarrow (\infty, 0) \]
\[ 1 \rightarrow (r_1, \infty) \]
\[ X \rightarrow (x, x) \]

where \( r_1 \) (or \( r_{1i} \)) is a real number which implies a possible discharging (or charging) strength. Besides, let OPAND, OPOR and OPINV in the set S correspond to AND, OR and INV in the set B, respectively. Then a Strength expression with the input signal strengths is compatibly derived, from the Boolean expression with input signal levels.

\[ (RVDD, RGND) \]

Fig. 3 Mappings between Level and Strength

The evaluation of the Strength expression results in a pair \((RVDD, RGND)\) of strength values. Each strength value is computed by following the algebraic operations and the \((RVDD, RGND)\) of strength values. Each strength value is \((RVDD, RGND)\) is an element, of which strength values of a finite \( r \), an unknown \( x \) and an infinity. The pair \((RVDD, RGND)\) is a signal strength of the output, of which the signal level is supposed to be equal to that evaluated by the Boolean algebra. In order to derive the signal level from the pair \((RVDD, RGND)\) of strength values, the mapping of \( O: S \rightarrow B \) is defined as follows:

\[ (r_i, \infty) \rightarrow 1 \]
\[ (\infty, r_i) \rightarrow 1 \]
\[ (x, x) \rightarrow X \]

where \( r_i \) (or \( r_{1i} \)) is a real number \( c (0, \infty) \)

These mappings are deduced from the following two observations; (1) if an input signal level is 1 (or 0), its corresponding static strength includes a finite (infinite) discharging value, (2) the operations of OPAND, and OPOR in S are defined by reflecting the connection of transistor networks corresponding to AND and OR in B, which is described in section I(A). Thus, the Continuous Strength Algebra (CSAL) is used to evaluate a Strength expression corresponding to a Boolean expression, and provides a signal strength \((RVDD, RGND)\) of the output for the Boolean expression.

C. Examples

As an example, consider an expression of \( \neg(A \cdot B + C) \) in the two sets S and B. The operations of \( \cdot \) and \(+\) denote, AND, OR and NOT in B; and OPAND, OPOR and OPINV in S, respectively. In order to simplify the computation, all \( r_0 \) and \( r_{1i} \) in the mapping I are assumed to be 1; I: \( 0 \rightarrow (\infty, 1) \), \( 1 \rightarrow (1, \infty) \).

Three examples are presented to illustrate the evaluation procedure for Boolean expressions by the CSAL and to compare them with the Boolean algebra:

\[ I: 0 \rightarrow (\infty, 1), \quad 1 \rightarrow (1, \infty) \] for all inputs,

With \((A, B, C) = (0, 0, 1)\),
\[ eB = (0 * 0 + 1) = 0 \]
\[ eS = (011 * 011 + 111) = 111 \]
\[ = ((011 / 2) + 11) = (1, \infty) \]
\[ = (\infty, 1) \]
\[ O: (\infty, 1) \rightarrow 0 \]

With \((A, B, C) = (1, 0, 0)\),
\[ eB = (1 * 0 + 1) = 1 \]
\[ eS = (011 * 011 + 111) = 111 \]
\[ = ((011 / 2) + 11) = (\infty, 1) \]
\[ = (\infty, 1) \]
\[ O: (2, \infty) \rightarrow 1 \]

With \((A, B, C) = (X, 1, 0)\),
\[ eB = (X1 + 0) = X \]
\[ eS = ((x, x) * 111 + 111) = ((x, x) + 111) = (x, x) \]
\[ = (x, x) \]
\[ O: (x, x) \rightarrow X \]

For any input combinations, the evaluations on S, \( eS \), yields the signal strengths which are matched with the signal levels on B, \( eB \), which are shown in Table 2.

V. Evaluation for Logical MOS Gates

We have shown that the internal transistor network of a logical MOS gate can be depicted by using a complementary Boolean expression and that the static strength of an internal node in the transistor network represents the signal at the node, in a satisfactory way. Furthermore, it has been shown that the Continuous Strength Algebra (CSAL) can handle a Boolean expression with a strength capability.

Based on these considerations, we have developed a new evaluation method for logical MOS gates. The output node of a logical MOS gate is represented by a static strength, depending on the following three factors; (1) the transistor network of the logical MOS gate, (2) the resistance of transistors lying in the transistor network, and (3) the inputs stimulating the transistor network. At first, we will associate the factors of (2) and (3) with the mapping I: B→S. Then, two evaluation methods are presented to derive the static strength from two kinds of logical MOS gate.

A. Input Strength Encoding

When an input stimulates the driver-load transistor network of a logical MOS gate, its signal level dictates the connectivity of an enhancement transistor in the driver network or in the load network. A 0-input transistor in the driver (load) network is turned into the off (on) state; and a 1-input transistor in the driver (load) networks is turned into the on (off) state. This implies that the 0(1)-input induces an infinite resistance in the driver (load) network and a finite on-channel resistance in the load (driver) network. An X-input transistor in the driver-load transistor network is characterized by an unknown resistance, ranging from finite on-channel resistance to infinity.
dependent on a number of factors, such as; transistor geometry (W/L), transistor type (p-type/n-type) and mode (enhancement/depletion), fabrication process (threshold voltages of n and p-type enhancement transistor, $V_{thn}$/$V_{thp}$), operating transistor-terminal voltages, etc. Normally, the on-channel of n and p-type enhancement transistor, $V_{thn}$/$V_{thp}$, operating (enhancement/depletion), fabrication process (threshold voltages transistors, existing in the driver-load network of a logical MOS gate, are different from each other in geometry, their on-channel resistances are estimated from their size and the design rule.

For an input $i$ of the logical MOS gate, a pair $(r_{pi}, r_{ni})$ of finite resistances, where $p$ and $n$ indicate p-type and n-type enhancement transistors, is given to represent the controlled two transistors in the load and driver networks. A 0 (or 1) signal level activates an infinite resistance at the load (driver) network, and a 0 (or 1) signal level activates an infinite resistance at the driver (load) network. Then, the pair of transistor-resistances controlled by the input is associated with the signal level through the mapping of $I : B \rightarrow S$ in the CSAL, such as:

<table>
<thead>
<tr>
<th>$A$</th>
<th>$B$</th>
<th>$C$</th>
<th>$r_{pi}$</th>
<th>$r_{ni}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>(5)</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>(6)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>(2)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>(0)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>(1)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>(2)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>(3)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>(4)</td>
</tr>
</tbody>
</table>

This mapping, denoted as an Input Strength Encoding, shows that an input signal level is encoded into an input signal strength for a logical MOS gate.

The input strength encoding is designed to evaluate the static strength at the output node of the logical MOS gate described by a complementary Boolean expression. In order to compute the static strength at the output by utilizing the CSAL, this mapping is naturally deduced by the following deliberations: (1) the transistor resistances in the driver and load networks are associated with the strength values of RVDD and RGND through the mapping $I : B \rightarrow S$ in the CSAL, (2) they are manipulated by the operations of OPAND and OPOR, which are derived from reflecting the correspondence of the transistor connections, in the driver and load networks, with the AND and OR operations, in the Boolean algebra and, then, the RVDD in the driver network and the RGND in the load network are derived, (3) the OPINV operation, corresponding to the INV operation in the complementary Boolean expression, interchanges the strength values to produce the RVDD in the load network and the RGND in the driver network. In (1) and (2), the driver (load) network is associated with the charging (discharging) strength value of RVDD (RGND). Actually, such reverse physical attributes of the networks are recovered in (3), and the result is physically matched with the static strength (SVDD,SGND) at the output node of the logical MOS gate.

### B. Evaluation for Static Gate

A static gate is uniquely represented by a Boolean expression. For the CMOS static gate (Figure 2(a)), consisting of enhancement transistors in the driver-load network, a complementary Boolean expression describes its internal transistor network. Inputs of the static gate can be translated into the signal strengths by the input strength encoding. For a Strength expression, compatible with the Boolean expression, the evaluation by using the CSAL is carried out to compute the static strength (SVDD,SGND) at the output node. The SVDD is a charging strength value in the load network and the SGND is a discharging strength value in the driver network. Since the two networks (i.e., driver and load) in the CMOS static gate have dual structures to each other, the evaluation results in three static strengths, namely; $(\infty, r)$, $(r, \infty)$ and $(x, x)$, where $r$ is a finite constant in the interval $(0, \infty)$. The result is translated into the signal level at the output node through the mapping of $O : S \rightarrow B$ in CMOS static gate, such as:

- $(\infty, r) \rightarrow 0$
- $(r, \infty) \rightarrow 1$
- $(x, x) \rightarrow X$

However, the nMOS static gate (Figure 2(b)) uses a single depletion transistor as the load network, and its Boolean expression only specifies the driver network but not the load network. This implies that the evaluation by utilizing the CSAL can compute a valid discharging strength value $SGND$ in the driver network, but cannot compute a valid charging strength value SVDD in the load network. Since the depletion transistor, which is independent of inputs, is always turned-on, its finite on-channel resistance, denoted as $r_{dep}$, becomes the charging strength value of the output node. Then, the signal strength at the output node is depicted by the static strength of a pair $(r_{dep}, SGND)$. Due to the single depletion transistor in the load network, the evaluations results in three static strengths; $(t_{dep}, r)$, $(t_{dep}, \infty)$ and $(t_{dep}, x)$. Then the pairs are translated into the signal levels at the output node through the mapping of $O : S \rightarrow B$ in nMOS static gate, such as:

- $(t_{dep}, r) \rightarrow 0$ when $R \leq \text{Lower limit}$
- $(t_{dep}, r) \rightarrow 1$ when $R \geq \text{Upper limit}$
- $(t_{dep}, \infty) \rightarrow X$ when Lower limit $\leq R \leq \text{Upper limit}$
- $(t_{dep}, x) \rightarrow X$

where $R = t(t_{dep} + r)$.

In Table 3, we show several evaluations for the CMOS and nMOS static gates of the example in section IV(C): $elg$ for CMOS is done with three pairs of finite resistances; $(r_{pa}, r_{na})$, $(r_{pb}, r_{nb})$ and $(r_{pc}, r_{nc})$, where $r_{ij}$ is the finite on-channel resistance of $i$-type enhancement transistor controlled by input $i$.

<table>
<thead>
<tr>
<th>$A$</th>
<th>$B$</th>
<th>$C$</th>
<th>$elg$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>(5)</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>(6)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>(2)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>(0)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>(1)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>(2)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>(3)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>(4)</td>
</tr>
</tbody>
</table>

For $elg$ for CMOS and $elg$ for nMOS, it is obvious that this evaluation method provides the static strength for the output node of the static gate by utilizing the CSAL.
A high impedance gate consists of a CMOS static gate, corresponding to a complementary Boolean expression, and two serially connected enable transistors. This configuration leads naturally to the derivation of two strength expressions from the Boolean expression, such as $e_{\text{load}} = -(e \oplus \Phi_p)$ and $e_{\text{driver}} = -(e \oplus \Phi_d)$, where $\Phi_p$ and $\Phi_d$ are two enable clock inputs. This derivation is due to the relationship between the operations and the transistor networks as regards the two enable transistors in the series connection with the driver and load networks.

The evaluation by utilizing the CSAL yields two pairs $(\text{SVDD}_{\text{load}}, \text{SGND}_{\text{load}})$ and $(\text{SVDD}_{\text{driver}}, \text{SGND}_{\text{driver}})$ or the static strengths at the output node. Since the SGND load and SVDDdriver are not matched with the physical attributes, such as the charging strength of the load network and the discharging of the driver network, the two pairs are deduced into a pair $(\text{SVDD}_{\text{load}}, \text{SGND}_{\text{driver}})$. This SVDDload and SGNDdriver become the charging and discharging strength values of the high impedance gate, respectively. Compared with the CMOS static gate, the evaluation results in the additional states of $(x,-)$, $(-,x)$ and $(-,-)$. The $(x,-)$ and $(-,x)$ implies that the X states are inclined to 1 and 0, respectively; and $(-,-)$ indicates a high impedance state $Z$. In the high impedance state, the signal strength of the dynamic strength is represented by the amount of charge in the output nodal capacitance, which is determined from its previous signal level. The signal strengths, derived from the above two evaluations, are translated to the signal levels through the mapping of $0: S \rightarrow B$ in high impedance gate, such as:

\[(\omega, \infty) \rightarrow 0 \]
\[(x, x) \rightarrow X \]
\[(x, \infty) \rightarrow X \]
\[(\omega, \omega) \rightarrow Z \]

As an example, we consider a high impedance gate comprising the CMOS static gate of $-(A \cdot B + C)$, as shown in Figure 1(c), and two enable clock inputs of $\Phi_p$ and $\Phi_d$. For the two enable clock inputs of $\Phi_p$ and $\Phi_d$, $(\Phi_p \Phi_d = 1,1)$ is presumed. The three static strengths of the CMOS static gate are combined with the enable clock inputs, which produces the static strengths of the high impedance gate, as illustrated in Table 4.

### Table 3 Evaluations for CMOS and nMOS Static Gates of $-(A \cdot B + C)$

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>$e_{\text{load}}$</th>
<th>$e_{\text{driver}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

### Table 4 Evaluations for High Impedance Gate of $-(A \cdot B + C)$

<table>
<thead>
<tr>
<th>$e_{\text{load}}$</th>
<th>$e_{\text{driver}}$</th>
<th>$e_{\text{load}}$</th>
<th>$e_{\text{driver}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

### VI. Comparisons with other approaches

We have proved that the new evaluation, utilizing the CSAL, computes the signal level and strength at the output node of a logical MOS gate. The new evaluation described here has several advantages in simulating the unidirectional logical primitive (i.e., logical MOS gate) in digital MOS circuits, over other evaluations such as: (1) gate-level evaluations with a limited discrete strength set $[2,3]$, (2) gate-level evaluation with a restricted continuous strength set $[4,5]$, (3) switch-level evaluation with a discrete strength set $[1]$. In order to assess the performance of the new evaluation method, we discuss how the other evaluations deal with a typical MOS gate to simulate the signal strength at the output node. An example is the nMOS static gate of $-(A \cdot B + C)$, which is shown in Figure 1(d). In Figure 4, four evaluation models are illustrated with the two input combinations of $(A,B,C)=(1,1,0)$ and $(0,0,1)$.
For the two input combinations, the gate-level evaluation yields a signal level at the output node 0, such as \(-(\text{A*B+C})=-1=0\). Then, in order to account for the signal strength at the output, its signal level is characterized using the discrete/continuous strength set. In the evaluations with a limited discrete strength set (Figure 4(a)), the signal level at the output node 0 is associated with the Driven strength, since the nMOS gate only delivers a Driven 0 because its drain network consists of enhancement transistors, or a Weak (or Resistive) Drive 1 because of its depletion load network. Then, we obtain an indistinguishable signal of Driven 0 for the two input combinations. As another more realistic approach at the gate-level, the evaluation with a restricted continuous strength set assigns one of two equivalent resistances \((R_I/R_P)\) to the signal strength at the output node. However, this approach also yields an indistinguishable signal of 0 with \(R_I\) strength for the two input combinations.

It is obvious that the gate-level evaluations do not accurately simulate the signal strength and they only work for a very restricted class of gate, such as an inverter. The higher abstraction of the gate-level inherently leads to incurring such a limitation in estimating the signal strength. Actually, the signal strength at the output node is varied with the transistor conduction paths between the power nodes and the output, which are settled by the gate inputs. This implies that the signal strength is not a function of the projected signal level at the output node, likewise in gate-level evaluations, but a function of the gate inputs and the internal transistor network implementing the gate.

In the switch-level evaluation, with a discrete strength set, two different channel-graphs, shown in Figure 4(c), are formed following the two input combinations. Then we formulate three matrices of blocking, charging, and discharging paths to find the strongest charging path \((u)\) and the strongest discharging path \((d)\) into every node in the channel-graphs. We could obtain the signal strengths, as follows; \((u_1, d_1)=(0,3)\), \((u_2, d_2)=(0,3)\), \((u_3, d_3)=(0,0)\) for \((A,B,C)=(1,1,0)\) and \((u_1, d_1)=(0,3)\), \((u_2, d_2)=(0,0)\), \((u_3, d_3)=(0,0)\) for \((A,B,C)=(0,0,1)\). These are derived from the signal levels at the nodes, such as; \(n_1=0\), \(n_2=0\), \(n_3=1\), \(n_4=0\) for \((1,1,0)\) and \(n_1=0\), \(n_2=0\), \(n_3=1\), \(n_4=0\) for \((0,0,1)\). It is noted that the conduction paths are taken into account to evaluate the signal strengths of the nodes. Although the switch-level evaluation simulates the signal strength at the output node as a function of the gate inputs and the transistor network, the evaluation results demonstrate a serious weakness in the evaluation of the signal strength. The node \(n_1\) has the same signal strength of \((u_1, d_1)=(0,3)\) for the two input combinations, in which typically two problems are found. One problem is that the results are not involved with the charging strength of the depletion transistor. This erroneous result, due to the evaluation strategy that one strength is able to override any number of weaker strengths, cannot even describe the behavior of a simple ratioed-circuit. The other problem is that they are indistinguishable from each other, despite their different external circuitry, which depends on the inputs. Normally, the discharging path through the two serial connected transistors (i.e., A and B) is supposed to be different from that through a single transistor (i.e., C). This shows a modeling deficiency in the evaluation of the signal strength. Furthermore, this evaluation has an advantage over the gate-level evaluations in computation time, since the formation of the channel-graph for each input combination incurs a severe computing overhead; and all nodes of \(n_1, n_2, n_3, n_4\) in the channel-graphs should be considered and evaluated for each event, redundantly.

In the new evaluation for the Boolean expression (Figure 4(d)), the signal strengths at the output node are represented by

\[
O = I(A,B,C) = A + B + C
\]

Given the values of \(t_{\text{dep}}=t_{\text{fa}}=t_{\text{fa}}=t_{\text{rc}}\) and the MOS circuit technique of nMOS

**Fig. 4 Typical evaluation models for an unidirectional logical primitive (i.e. logical MOS gate)**

\((t_{\text{dep}}, t_{\text{fa}}, t_{\text{ra}}, \text{B} \text{ and } t_{\text{dep}}, t_{\text{rc}})\) for the two input combinations of \((A,B,C)=(1,1,0)\) and \((0,0,1)\), respectively. The results demonstrate that the signal levels at the output node are dictated by which strength is stronger to the output node; and that the input has two different signal strengths depending on the two input combinations. It is obvious that the new evaluation method, utilizing the CSAL, offers reasonable continuous strength values to the output node.

Based on this discussion, we compare the new evaluation with the other evaluations, at the gate-level and the switch-level, from the standpoint of accuracy and computation. In Table 5: This table shows that the new evaluation delivers performance accuracy at the expenditure of gate-level computation. In simulating the unidirectional logical primitive (modeled as the logical MOS gate here), the new evaluation utilizing the CSAL, has several advantages over other evaluations, namely; it simulates the strength at the output node as a function of gate inputs and the internal transistor network, it evaluates a complementary Boolean expression at the gate-level, it provides evaluation accuracy of the switch-level, it computes more realistic physical entities (i.e., two variable equivalent resistances from the gate output to the power node.)

**VIII. Timing model for logical MOS gates**

In the new evaluation, utilizing the CSAL, the signal strength of \((SVDD, SGND)\) at the output node consists of the charging and discharging strengths from the power nodes to the output node. In fact, they are the two equivalent resistances from the gate output node, with respect to the VDD and GND power node, as shown in Figure 5(a). These resistances form a basis to model the output impedance in Thevenin's circuit [4]. The output impedance depends on the signal level and the static strength of \((SVDD, SGND)\) in the following: If the 1 (or 0) signal level is projected to the output node, the SVDD (or SGND) becomes the output impedance of the gate; otherwise (i.e., X signal level), the parallel addition of \(SVDD/2\) (i.e., \(SVDD + X SGND(SVDD + SGND)\)) is assigned to the output impedance. Since the strength value is the function of the gate inputs and the transistor network, the output impedance is varied with the gate inputs and the transistor network, as shown in Figure 5(b).

With the output impedance, a timing model for logical MOS gates, based on a simple RC model, has been developed. The gate delay is defined as a product of the output impedance and the output nodal capacitance. Then, the rising/falling/unknown delay can be expressed as:

\[
T_d = SVDD \times C
\]

for the rising delay.
for the falling delay
T_f = (SVDD|SGND) × C
for the unknown delay
where C represents the output nodal capacitance.

With the delay model, we consider an example of -(A*B+C) shown in Figure 6(a). For the two input combinations of (A,B,C) = (0,0,1) and (1,0,0), the falling delay is computed as
T_f = SGND × C = C (Figure 6(b)); and the rising delay becomes
T_r = SVDD × C = 2C (Figure 6(c)). These show that the rising delay is two times larger than the falling delay.

Most timing models at the gate-level, are faced with an inherent limitation of a single (rising/falling) delay, which is due to using a higher logical abstraction. For the logical MOS gate with more than one input, the single delay does not suffice to describe different delays from the inputs to the output (i.e., multiple inputs-to-outputs delays). However, it is shown that those multiple inputs-to-output delays could be reasonably handled by the timing model using the static strength (SVDD,SGND).

VIII. Conclusions

In this paper, we began by modeling a unidirectional logical primitive as a logical MOS gate, having the following features (1) multiple inputs and a single output, (2) a driver-load configuration, and (3) a static Boolean logic function. At the same time, we have shown that an internal configuration of a transistor network, in a logical MOS gate, can be described by a complementary Boolean expression. We, then, introduced a new resistive and capacitive signal model consisting of static and dynamic signal strengths, which accurately describe the signal at a node. Then, we developed a new Continuous Strength ALgebra (CSAL) to handle Boolean expressions with a strength capability. Following these discussions, we proposed a new evaluation for logical MOS gates utilizing the CSAL, while proving that this approach offers the signal level and strength at the output node of the logical MOS gate described by a complementary Boolean expression. In order to assess its performance in simulating a unidirectional logical primitive (i.e., the logical MOS gate), we compared the new evaluation with the other evaluations and demonstrated its advantages. Finally, we discussed a timing model for logical MOS gates, which is based on a simple RC model, and demonstrated that the new evaluation reasonably provides the multiple inputs-to-output delays for the gate output, while handling the output impedance in the Thevenin's circuit with a function of the gate inputs and the transistor network implementing the gate.

The goal of this research was to find a simple and efficient switch-level evaluation method for the unidirectional logical primitives in a mixed-level (i.e., gate and switch) simulation of digital MOS circuits. We have demonstrated that the evaluation utilizing the Continuous Strength ALgebra (CSAL) fulfills this goal by offering switch-level accuracy at the cost of gate-level computation.


<table>
<thead>
<tr>
<th>Evaluation</th>
<th>Gate-level</th>
<th>New</th>
<th>Switch-level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logic level</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Evaluation</th>
<th>Gate-level</th>
<th>New</th>
<th>Switch-level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strength</td>
<td>Poor</td>
<td>Accurate</td>
<td>Moderate</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Evaluation</th>
<th>Gate-level</th>
<th>New</th>
<th>Switch-level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expression</td>
<td>Boolean eq.</td>
<td>Boolean eq.</td>
<td>Matrix</td>
</tr>
<tr>
<td>Operation</td>
<td>Binary</td>
<td>Serial &amp; Parallel Addition</td>
<td>Mix./Max.</td>
</tr>
<tr>
<td>Number</td>
<td>Finite Integer</td>
<td>Real</td>
<td>Finite Integer</td>
</tr>
<tr>
<td>Iteration</td>
<td>1</td>
<td>1</td>
<td>r (Max. number of connection)</td>
</tr>
<tr>
<td>Computation</td>
<td>Order (n of input)</td>
<td>N</td>
<td>k × N</td>
</tr>
</tbody>
</table>

Tab.5 Comparisons in Accuracy and Computation